Learning Automata-based Solutions to the Optimal Web Polling Problem Modelled as a Nonlinear Fractional Knapsack Problem*

Ole-Christoffer Granmo† and B. John Oommen‡

Abstract

We consider the problem of polling web pages as a strategy for monitoring the world wide web. The problem consists of repeatedly polling a selection of web pages so that changes that occur over time are detected. In particular, we consider the case where we are constrained to poll a *maximum* number of web pages per unit of time, and this constraint is typically dictated by the governing communication bandwidth, and by the speed limitations associated with the processing. Since only a fraction of the web pages can be polled within a given unit of time, the issue at stake is one of determining which web pages are to be polled, and we attempt to do it in a manner that maximizes the number of changes detected. We solve the problem by first modelling it as a *Stochastic Non-linear Fractional Knapsack Problem*. We then present an on-line Learning Automata (LA) system, namely, the *Hierarchy of Twofold Resource Allocation Automata* (H-TRAA), whose primitive component is a *Twofold Resource Allocation Automaton* (TRAA). Both the TRAA and the H-TRAA have been proven to be asymptotically optimal. Finally, we demonstrate empirically that the H-TRAA provides *orders of magnitude* faster convergence compared to the Learning Automata Knapsack Game (LAKG) which represents the state-of-the-art for this problem. Further, in contrast to the LAKG, the H-TRAA scales sub-linearly. Based on these results, we believe that the H-TRAA has also tremendous potential to handle demanding real-world applications, particularly those which deal with the world wide web.

Keywords: *Web Polling, WWW-related Resource Allocation, Nonlinear Knapsack Problems, Hierarchical Learning, Learning Automata, Stochastic Optimization*

*The second author was partially supported by NSERC, the Natural Sciences and Engineering Research Council of Canada. A preliminary version of this paper was presented at IEA/AIE'08, the 2008 International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, Wroclaw, Poland, in June 2008.

†Author's status: *Associate Professor*. This author can be contacted at: Department of Information and Communication Technology, University of Agder, Grooseveien 36, NO-4876 Grimstad, Norway. E-mail: ole.granmo@uia.no.

‡Author's status: *Chancellor's Professor*, *Fellow of the IEEE*, and *Fellow of the IAPR*. This author can be contacted at: School of Computer Science, Carleton University, Ottawa, Canada : K1S 5B6. The author is also an Adjunct Professor with the University of Agder in Grimstad, Norway. E-mail: oommen@scs.carleton.ca.

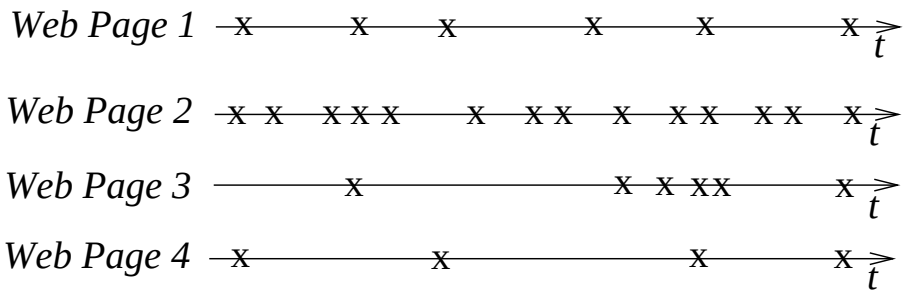

Figure 1: Web page changes occurring over time. An 'x' on the time-lines denotes that the respective web page has changed. Observe that the occurrence of this event is not observable to the outside world unless the web page is polled. Observe too that the rates at which the various web pages change are not the same, implying that the corresponding polling frequencies should also be page-dependent.

1 Introduction

The world wide web is an extremely vast resource-thirsty field, which probably consumes a major portion of the computing resources available today. Searching, updating and examining web-pages is, undoubtedly, one of the primary tasks done by both individuals and companies today. This, in turn, leads to numerous extremely interesting real-life resource allocation and scheduling problems, and in this paper, we study one such problem, the so-called "Web polling" problem.

Web page monitoring consists of repeatedly polling a selection of web pages so that the user can detect changes that occur over time. Clearly, as this task can be prohibitively expensive, in practical applications, the system imposes a constraint on the *maximum* number of web pages that can be polled per time unit. This bound is dictated by the governing communication bandwidth, and by the speed limitations associated with the processing. Since only a fraction of the web pages can be polled within a given unit of time, the problem which the system's analyst encounters is one of determining which web pages are to be polled. In such cases, a reasonable choice of action is to choose web pages in a manner that maximizes the number of changes detected, and the optimal allocation of the resources involves trial-and-failure. As illustrated in Figure 1, web pages may change with varying frequencies (that are unknown to the decision maker), and changes appear more-or-less randomly. Furthermore, as argued elsewhere, [6–8], the probability that an individual web page poll uncovers a change on its own, decreases monotonically with the polling frequency used for that web page.

Although several nonlinear criterion functions for measuring web monitoring performance have been proposed in the literature (e.g., see [14, 21]), from a broader viewpoint they are mainly built around the basic concept of the *update detection probability*, i.e., the probability that polling a web page results in new information being discovered. Therefore, for the purpose of clarification and for the sake of conceptual clarity, we will use the update detection probability as the token of interest in this paper. To further define our notion of web monitoring performance, we consider that time is discrete with the time interval length, T, to be the atomic unit of decision making. In each time interval every single web page i has a constant

probability q_i of remaining *unchanged*[1]. Furthermore, when a web page is updated/changed, the update is available for detection only until the web page is updated again. Subsequent to that event, the original update is considered to be "lost". For instance, whenever a newspaper web page is updated, the previous news items are replaced by the most recent ones.

In the following, we will denote the update detection probability of a web page i as d_i. Under the above conditions, d_i depends on the frequency, x_i, that the page is polled with, and is modeled using the following expression:

$$d_i(x_i) = 1 - {q_i}^{\frac{1}{x_i}}.$$

By way of example, consider the scenario that a web page remains unchanged in any single time step with probability 0.5. Then, polling the web page uncovers new information with probability $1 - 0.5^3 = 0.875$ if the web page is polled every 3^{rd} time step (i.e., with frequency $\frac{1}{3}$) and $1 - 0.5^2 = 0.75$ if the web page is polled every 2^{nd} time step. As one can observe, increasing the polling frequency reduces the probability of discovering new information on each polling.

A number of classical policies for the Web Polling Problem have been discussed in the literature (see e.g., [14]). The *Uniform* policy allocates monitoring resources uniformly across all web pages. This classical policy can be applied directly in an unknown environment. In the so-called *Proportional* policy, the allocation of monitoring resources to web pages is proportional to the update frequencies of the web pages. Accordingly, this policy requires that the web page update frequencies are known. The *Estimator* policy, on the other hand, handles unknown web update frequencies by polling web pages *uniformly* in a parameter estimation phase, with the purpose of estimating the update frequencies. After the parameter estimation phase, the scheme applies the proportional policy, where, however, the latter is based on the *estimated* update frequencies rather than the true ones. Finally, the *Optimal* policy requires that the web page update frequencies are known, and determines the optimal solution based on the principle of Lagrange multipliers [14,21].

Although the Web Polling problem is quite general, we shall proceed to solve it by suggesting that it falls within the model of knapsack-based problems. Indeed, in the most general setting, we shall utilize the model of the Stochastic Non-linear Fractional Equality Knapsack (NEFK) problems to model the present problem, and once such a formalization has been established, we shall allude to the Learning Automata (LA) [11,18] solution of the NEFK problem, proposed in [9], to solve the Web Polling problem. LA have previously been used to model biological systems [19], and have attracted considerable interest in the last decade because they can learn the optimal actions when operating in (or interacting with) unknown stochastic environments. Furthermore, they combine rapid and accurate convergence with low computational complexity.

The novel Learning Automata Knapsack Game (LAKG) scheme that we proposed in [8] does not rely on estimating parameters, and can be used to solve the Stochastic NEFK problem in both static and dynamic

[1]Note that in our empirical results, we also report a high monitoring performance even with changing q_i. The high performance can be explained by the ability of our scheme to achieve adaptation.

settings. Indeed, empirical results verify that the LAKG finds the optimal solution with arbitrary accuracy, guided by the principle of Lagrange Multipliers. Furthermore, the empirical results show that the performance of the LAKG is superior to that of parameter-estimation-based schemes, both in static and dynamic environments. Accordingly, we believe that the LAKG can be considered to represent the state-of-the-art when it concerns research on the stochastic NEFK problem. This landmark is now extended to develop the *Twofold Resource Allocation Automaton* (TRAA), (which, in itself is the first reported LA which is *artificially* ergodic[2]), and its hierarchical version, the H-TRAA – which is the strategy used to solve the problem being investigated.

1.1 Contributions of This Paper

The contributions of this paper are the following:

1. We report the first *analytical* results for schemes that solve the optimal Web Polling problem using a formal solution to the Stochastic NEFK problem.

2. The solution we propose involves a novel scheme for the *two-material* resource allocation problem, namely, the *Twofold Resource Allocation Automaton (TRAA)*, which is the first reported LA that is *artificially* rendered ergodic, and which is proven to be asymptotically optimal[3].

3. The solution we propose for web-polling also utilizes the H-TRAA, which is the first *hierarchical* solution to the Stochastic NEFK Problem, based on a hierarchy of TRAAs.

4. We verify empirically that the H-TRAA provides orders of magnitude faster convergence compared to the LAKG for the web-polling problem.

As a result of the above contributions, we believe that the H-TRAA is the first reported viable and realistic strategy for solving the optimal Web Polling problem. Indeed, it can also be used for other optimal allocation of sampling resources in large scale web accessibility assessment [16].

1.2 Paper Organization

The paper is organized as follows. In Section 2, we formulate the targeted problem formally and discuss state-of-the-art solutions. Then, in Section 3 we present the Twofold Resource Allocation Automaton (TRAA) for the *two-material* problems, and prove its asymptotic optimality. We continue with proposing how TRAAs can be arranged in a hierarchy for solving *multi-material* Stochastic NEFK Problems, and in Section 4, verify empirically that the H-TRAA provides orders of magnitude faster convergence compared

[2]LA which have been artificially made *absorbing* to yield specific properties, have been earlier reported [12]. However, we are not aware of any LA which, in essence are absorbing, but which have been made artificially *ergodic*.

[3]As we shall clarify later, the formal results about the design and convergence of the TRAA and H-TRAA are also found elsewhere [9]. We have included them here just for the sake of completeness, so that this paper can be a stand-alone publication. This was also the recommendation of the Referees. We emphasize, though, that the experimental results in [9] do *not* pertain to the application domain studied in this paper.

to the LAKG when applied to the optimal Web Polling problem. Indeed, we shall present results that clearly demonstrate that the H-TRAA allows us to tackle $32,768$-parameter problems in *real-time*. The solution also permits the system to be dynamic! Finally, we offer suggestions for further work before we conclude the paper in Section 5.

2 The Stochastic Non-linear Equality Fractional Knapsack Problem

We first formulate, in a fairly general setting, a set of knapsack-based problems that are, in actuality, related to the web-polling problem. In a multitude of real-world situations, resources are often to be allocated based on incomplete and noisy information. Such resource allocation problems are particularly intriguing because, in many cases, incomplete and noisy information render traditional optimization techniques ineffective. In this paper, we address one such model which can be translated into a family of problems:

> *Imagine that you have to allocate a limited amount of time among n different activities. The problem is such that spending a time instant on an activity randomly produces one of two possible outcomes — the time instant is either spent "fruitfully" or "unfruitfully". In this generic setting, your goal is to maximize the expected amount of fruitfully spent time. Unfortunately, you are only given the following information regarding the activities:*
>
> 1. *Each instant of time spent on an activity has a certain probability of being fruitful, and*
> 2. *This probability decreases with the amount of time spent on the activity.*
>
> *To render the problem even more realistic, you do not have access to the probabilities themselves. Instead, you must rely on solving the problem by means of trial-and-failure, i.e., by attempting different allocations, and observing the resulting random outcomes*[4].

The above problem instances can be formulated as a *Stochastic Non-linear Fractional Equality Knapsack (NEFK) Problems* as exemplified earlier [6–8]. Such a formulation permits an analytically rigorous treatment of the problem.

2.1 Classical Linear and Nonlinear Fractional Knapsack Problems

In order to appreciate the qualities of the Stochastic NEFK Problem, it is beneficial to view the problem in light of the classical *linear* Fractional Knapsack (FK) Problem. Indeed, the Stochastic NEFK Problem generalizes the latter problem in two significant ways. Both of the two problems are *briefly* defined below.

[4]Students frequently encounter a version of the above intriguing problem. A student that pursues several different topics in a semester has to decide how to allocate his working hours among the topics. After a day of work, the student will have some idea of how much he has learned during the day, allowing him to assess his current allocation of working hours. Rather than over specializing in a single topic and treating the other topics superficially, seeking overall mastery of the topics could be a wise choice in this situation. However, the amount of time required to master a topic will vary, simply because the nature of a topic influences the student's "learning curve" for that topic. Thus, finding an optimal allocation in this problem must involve trial and failure, and unknown success probabilities, as in our present model. *Thus, effectively, we are treating each mechanism which has to determine the web-polling frequency, as a "student".*

The Linear Fractional Knapsack (FK) Problem: The linear FK problem is a classical continuous optimization problem which also has applications within the field of resource allocation. The problem involves n materials of different value v_i per unit volume, $1 \leq i \leq n$, where each material is available in a certain amount $x_i \leq b_i$. Let $f_i(x_i)$ denote the value of the amount x_i of material i, i.e., $f_i(x_i) = v_i x_i$. The problem is to fill a knapsack of fixed volume c with the material mix $\vec{x} = [x_1, \ldots, x_n]$ to yield a maximal value for $\sum_1^n f_i(x_i)$ [1].

The Nonlinear Equality FK (NEFK) Problem: One important extension of the above classical problem is the *Nonlinear Equality* FK problem with a separable and concave objective function. The problem can be stated as follows [10]:

$$\begin{array}{ll} \text{maximize} & f(\vec{x}) = \sum_1^n f_i(x_i) \\ \text{subject to} & \sum_1^n x_i = c \text{ and } \forall i \in \{1, \ldots, n\}, x_i \geq 0. \end{array}$$

Since the objective function is considered to be concave, the value function $f_i(x_i)$ of each material is also concave. This means that the derivatives of the material value functions $f_i(x_i)$ with respect to x_i, (hereafter denoted f_i'), are non-increasing. In other words, the material value *per unit volume* is no longer constant as in the linear case, but decreases with the material amount, and so the optimization problem becomes:

$$\begin{array}{ll} \text{maximize} & f(\vec{x}) = \sum_1^n f_i(x_i), \text{where } f_i(x_i) = \int_0^{x_i} f_i'(x_i) dx_i \\ \text{subject to} & \sum_1^n x_i = c \text{ and } \forall i \in \{1, \ldots, n\}, x_i \geq 0. \end{array}$$

Efficient solutions to the latter problem, based on the principle of Lagrange multipliers, have been devised. In short, the optimal value occurs when the derivatives f_i' of the material value functions are equal, subject to the knapsack constraints [2, 5]:

$$\begin{array}{l} f_1'(x_1) = \cdots = f_n'(x_n) \\ \sum_1^n x_i = c \text{ and } \forall i \in \{1, \ldots, n\}, x_i \geq 0. \end{array}$$

2.2 The Stochastic NEFK Problem

In this paper we generalize the above nonlinear equality knapsack problem. First of all, we let the material value per unit volume for any x_i be a *probability* function $p_i(x_i)$. Furthermore, we consider the distribution of $p_i(x_i)$ to be *unknown*. That is, each time an amount x_i of material i is placed in the knapsack, we are only allowed to observe an instantiation of $p_i(x_i)$ at x_i, and not $p_i(x_i)$ itself. Given this stochastic environment, we intend to devise an on-line incremental scheme that learns the mix of materials of maximal *expected* value, through a series of informed guesses. Thus, to clarify issues, we are provided with a knapsack of fixed volume c, which is to be filled with a mix of n different materials. However, unlike the NEFK, in the Stochastic NEFK Problem the unit volume value of a material i, $1 \leq i \leq n$, is a random quantity — it takes the value 1 with probability $p_i(x_i)$ and the value 0 with probability $1 - p_i(x_i)$, respectively. As an additional complication, $p_i(x_i)$ is nonlinear in the sense that it decreases monotonically with x_i, i.e.,

$x_{i_1} \leq x_{i_2} \Leftrightarrow p_i(x_{i_1}) \geq p_i(x_{i_2})$.

Since unit volume values are random, we operate with the expected unit volume values rather than the actual unit volume values themselves. With this understanding, and the above perspective in mind, the expected value of the amount x_i of material i, $1 \leq i \leq n$, becomes $f_i(x_i) = \int_0^{x_i} p_i(u)du$. Accordingly, the expected value per unit volume[5] of material i becomes $f_i'(x_i) = p_i(x_i)$. In this stochastic and non-linear version of the FK problem, the goal is to fill the knapsack so that the expected value $f(\vec{x}) = \sum_1^n f_i(x_i)$ of the material mix contained in the knapsack is maximized as below:

$$\begin{array}{ll} \text{maximize} & f(\vec{x}) = \sum_1^n f_i(x_i), \text{where } f_i(x_i) = \int_0^{x_i} p_i(u)du, \text{ and } p_i(x_i) = f_i'(x_i), \\ \text{subject to} & \sum_1^n x_i = c \text{ and } \forall i \in \{1, \ldots, n\}, x_i \geq 0. \end{array}$$

A fascinating property of the above problem is that the amount of information available to the decision maker is limited — the decision maker is only allowed to observe the current unit value of each material (either 0 or 1). That is, each time a material mix is placed in the knapsack, the unit value of each material is provided to the decision maker. The actual outcome probabilities $p_i(x_i), 1 \leq i \leq n$, however, remain *unknown*. As a result of the latter, the expected value of the material mix must be maximized by means of trial-and-error, i.e., by experimenting with different material mixes and by observing the resulting random unit value outcomes.

We conclude this section by stating that given the above considerations, we shall show that our aim is to find the page polling frequencies $\vec{x}$ that maximize the expected number of pollings uncovering new information per time step:

$$\begin{array}{ll} \text{maximize} & \sum_1^n x_i \times d_i(x_i) \\ \text{subject to} & \sum_1^n x_i = c \text{ and } \forall i = 1, \ldots, n, x_i \geq 0. \end{array}$$

Note that in the general web monitoring case, we are not able to observe $d_i(x_i)$ or q_i directly — polling a web page only reveals whether the web page has been updated *at least once* since our last poll[6]. As such, web monitoring forms a proof-of-concept application for resource allocation in unknown stochastic environments.

2.3 State-of-the-Art: The Stochastic NEFK problem

To the best of our knowledge, prior to our work reported in [8], the stochastic NEFK problem was not addressed in the literature before. However, several studies on related problems have been reported. For example, the works of [4, 17] consider solution policies for stochastic generalizations of the so-called NP-hard *linear* integer knapsack problem. In these papers, value distributions were considered known and

[5]We hereafter use $f_i'(x_i)$ to denote the derivative of the expected value function $f_i(x_i)$ with respect to x_i.

[6]Some web pages are also annotated with the time of last update. However, this information is not generally available/reliable [3], and is therefore ignored in our scheme.

constant, making dynamic programming a viable solution. Another variant of the knapsack problem is found in [15] where a deterministic knapsack is used, however, with objects arriving to and departing from the knapsack at random times. The optimization problem considered was to accept/block arriving objects so that the average value of the knapsack is maximized.

The first reported generic treatment of the stochastic NEFK problem itself can be found in [8]. The approaches that represent the non-LA state-of-the-art, assume that the knapsack problem is *deterministic* and fully *known*. However, from a web monitoring perspective the web must often be seen as a stochastic and more or less unknown environment. Various instantiations of the problem have, however, appeared sporadically, particularly within the web monitoring domain. In these latter instantiations, the unknown parameters of the knapsack problem are *estimated* by means of a tracking phase where web pages are polled mainly for estimation purposes [14, 21]. One major disadvantage of such an approach is that the parameter estimation phase significantly delays the implementation of an optimal solution. This disadvantage is further aggravated in *dynamic* environments where the optimal solution changes over time, introducing the need for parameter re-estimation [6].

With regard to the particular application domain, recent approaches to resource allocation in web monitoring attempt to *optimize* the performance of the system when the monitoring capacity is restricted [14, 21]. The principle cited in the literature essentially invokes Lagrange multipliers to solve a *nonlinear equality* knapsack problem with a separable and concave objective function [10]. Thus, for example, a basic web monitoring resource allocation problem may involve n web pages that are updated periodically, although with different periods. Clearly, each web page can be polled with a maximum frequency - which would result in a sluggish system. The problem which we study involves determining the web page polling frequencies (i.e., how often each web page is accessed by the monitoring system) so as to maximize the number of web page updates detected. Observe that this must be achieved without exceeding the available monitoring capacity — e.g., the maximum number of web pages that can be accessed per unit of time as dictated by the governing communication bandwidth and processing speed limitations.

2.4 Related Solutions to Knapsack-family Problems

In order to put our work in the right perspective, we first provide a brief review of the concepts and the solution found in [8] - which are also relevant for more "primitive" variants of the knapsack problem.

As indicated in the introduction, solving the classical linear FK problem involves finding the most valuable mix $\bar{x}^* = [x_1^*, \ldots, x_n^*]$ of n materials that fits within a knapsack of fixed capacity c. The material value per unit volume for each material i is given as a constant v_i, and each material is available in a certain amount $x_i \leq b_i$, $1 \leq i \leq n$. Accordingly, the value of the amount x_i of material i, $f_i(x_i) = v_i x_i$, is linear with respect to x_i. In other words, the derivative of $f_i(x_i)$ — i.e., the material value per unit volume — is fixed: $f_i'(x_i) = v_i$. Because a fraction of each material can be placed in the knapsack, the following greedy algorithm from [1] finds the most valuable mix: *Take as much as possible of the material that is most valuable per*

unit volume. If there is still room, take as much as possible of the next most valuable material. Continue until the knapsack is full.

Let us now generalize this and assume that the material unit volume values are *random* variables with *constant* and *known* distributions. Furthermore, for the sake of conceptual clarity, let us only consider binary variables that *either* instantiate to the values of 0 or 1. Since the unit volume values are random, let p_i denote the probability of the unit volume value $v_i = 1$ for material i, $1 \leq i \leq n$, which means that the probability of the unit volume value $v_i = 0$ becomes $1 - p_i$. With some insight, it becomes evident that under such conditions, the above greedy strategy can again be used to maximize the *expected* value of the knapsack, simply by selecting the material based on the *expected* unit volume values, $E[v_i] = 0 \times (1 - p_i) + 1 \times p_i$, rather than actual unit volume values.

The above indicated solution is, of course, inadequate when the p_i's are unknown. Furthermore, the problem becomes even more challenging when the p_i's are no longer constant, but rather depend on their respective material amounts x_i, $1 \leq i \leq n$. Let $p_i(x_i)$ denote the probability that the current unit volume value of material i is $v_i = 1$, given that the amount x_i has already been placed in the knapsack. Then, the expected value per unit volume of material i, $1 \leq i \leq n$, becomes $E[v_i] = 0 \times [1 - p_i(x_i)] + 1 \times p_i(x_i) = p_i(x_i)$, and accordingly, the expected value of the amount x_i becomes $f_i(x_i) = \int_0^{x_i} p_i(u)du$.

Our aim, then, is to find a scheme that moves towards optimizing the following NEFK problem on-line:

$$\begin{array}{ll} \text{maximize} & f(\vec{x}) = \sum_1^n f_i(x_i), \text{where} f_i(x_i) = \int_0^{x_i} p_i(u)du, \text{ and } p_i(x_i) = f_i'(x_i), \\ \text{subject to} & \sum_1^n x_i = c \text{ and } \forall i \in \{1, \ldots, n\}, x_i \geq 0. \end{array}$$

Note that we allow only instantiations of the material values per unit volume to be observed. That is, each time an amount x_i of material i is placed in the knapsack, an instantiation v_i at x_i is observed.

Because of the above intricacies, in [8] and in this present paper, we choose to approach the problem by relying on informed material mix *guesses*, i.e., by experimenting with different material mixes and learning from the resulting random unit volume value outcomes. We shall assume that x_i is any number in the interval $(0, 1)$. The crucial issue that we have to address, then, is that of determining how to change our current guesses on x_i, $1 \leq i \leq n$. . We shall attempt to do this in a discretized manner by subdividing the unit interval into N points $\{\frac{1^\lambda}{N^\lambda}, \frac{2^\lambda}{N^\lambda}, \ldots, \frac{(N-1)^\lambda}{N^\lambda}, 1\}$, where N is the resolution of the learning scheme and $\lambda > 0$ determines the linearity of the discretized solution space[7]. We will see that a larger value of N will ultimately imply a more accurate solution to the knapsack problem.

At this juncture, it is pertinent to mention that although the rationale for this updating is the stochastic point location solution proposed by Oommen in [13], the two schemes are quite distinct for the following reasons:

1. The method proposed in [13] assumes the existence of an Oracle which informs the LA whether to go "right" or "left". In our application domain, this now has to be *inferred* by the system.

[7]The importance of this parameter can be seen from the empirical results of [8]. In this paper, we have chosen to set λ to unity.

2. The method proposed in [13] assumes that there is only a single LA in the picture. Here, we specifically understand that there are multiple LAs organized in a hierarchy — each of them being constrained to work together with the others[8].

3. In [13] the problem of analyzing scenarios with space varying responses from the environment was left open. This problem is tackled in [8].

4. As opposed to the scheme in [13], our present approach is also applicable to dynamic (time varying) environments.

5. There is a "huge" fundamental difference between the LA which we devise here and the work of [13]. Unlike the latter, in which the system is truly ergodic, our present LA would be *absorbing* if the end-states of the probability space are also included. However, to forcefully render this present machine ergodic, we have artificially made the LA ergodic by *excluding* these states from the set of possible probability values. This makes the analysis both distinct and quite fascinating. As mentioned earlier, we are not aware of any LA which, in essence are absorbing, but which have been made artificially *ergodic*.

3 A Hierarchy of Twofold Resource Allocation Automata (H-TRAA)

3.1 Details of the TRAA Solution

3.1.1 Design of the TRAA Solution

We first present[9] our LA based solution to *two-material* Stochastic NEFK Problems. The two-material solution forms a critical part of the hierarchic scheme for multiple materials that is presented subsequently. As illustrated in Figure 2, our solution to two-material problems constitutes of three modules:

1. A Stochastic Environment

2. The TRAA itself, and

3. An Earliest Deadline First (EDF) Scheduler.

We first detail each of the three modules, before we analyze the overall *feedback connection* between them. Finally, we prove that the TRAA that we have developed in this section is asymptotically optimal for two-material Stochastic NEFK problems.

[8] It is conceivable that this problem can be resolved with a single LA possessing an extended number of actions. But we do not recommend it for scalability reasons — the action space would grow exponentially.

[9] As mentioned earlier, the formal results about the design and convergence of the TRAA and H-TRAA are also found elsewhere [9]. We have included them here just for the sake of completeness, so that this paper can be a stand-alone publication.

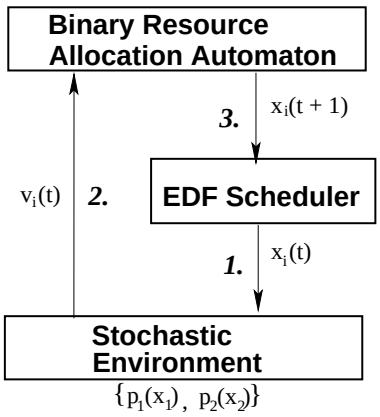

Figure 2: The Twofold Resource Allocation Automaton (TRAA) interacting with a Scheduler and an unknown Stochastic Environment.

Stochastic Environment: The *Stochastic Environment* for the two-material case can be characterized by:

1. The capacity c of the knapsack;

2. Two material unit volume value probability functions $p_1(x_1)$ and $p_2(x_2)$.

In brief, if the amount x_i of material i is suggested to the Stochastic Environment, the Environment replies with a unit volume value $v_i = 1$ with probability $p_i(x_i)$ and a unit volume value $v_i = 0$ with probability $1 - p_i(x_i)$, $i \in \{1, 2\}$. It should be emphasized that to render the problem both interesting and non-trivial, we assume that $p_i(x_i)$ is unknown to the TRAA.

Twofold Resource Allocation Automaton (TRAA): The scheme which attempts to learn the optimal allocation $\vec{x}^* = [x_1^*, x_2^*]$ can be described as follows. A finite fixed structure automaton with the states $s(t) \in \{1, 2, \ldots, N\}$ is used to decide the allocation of resources among the two materials. Let the current state of the automaton be $s(t)$. Furthermore, let $q_{s(t)}$ refer to the fraction $\frac{s(t)}{N+1}$, and let $r_{s(t)}$ refer to the fraction: $1 - q_{s(t)}$. Then the automaton's current guess is $\vec{x} = [q_{s(t)}, r_{s(t)}]$.

If the Stochastic Environment tells the automaton that the unit volume value of material i is $v_i(t)$ at time t, the automaton updates its state as follows:

$$s(t+1) := s(t) + 1 \quad \textbf{If rand}() \leq r_{s(t)} \textbf{ and } v_i(t) = 1 \textbf{ and } 1 \leq s_i(t) < N \textbf{ and } i = 1 \tag{1}$$

$$s(t+1) := s(t) - 1 \quad \textbf{If rand}() \leq q_{s(t)} \textbf{ and } v_i(t) = 1 \textbf{ and } 1 < s_i(t) \leq N \textbf{ and } i = 2 \tag{2}$$

$$s(t+1) := s(t) \qquad \textbf{Otherwise}. \tag{3}$$

Figure 3 shows the resulting stochastic transition graphs for resolution $N = 5$. The upper graph shows the transitions for feedback from the Stochastic Environment on material 1, and the graph below shows the transitions for feedback on material 2. Notice how the stochastic state transitions are designed to offset the

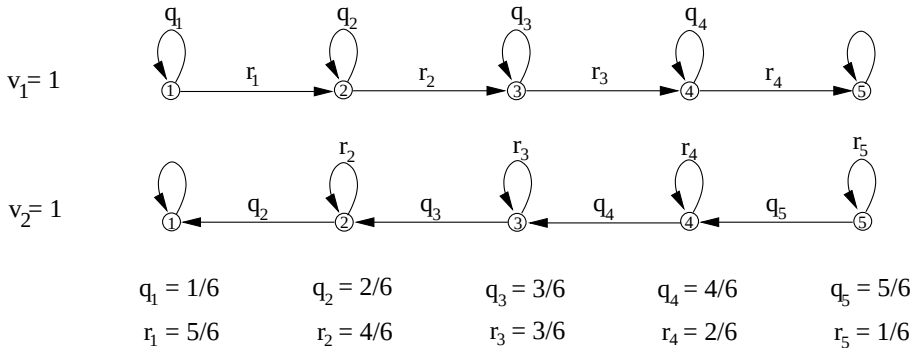

Figure 3: The stochastic transition graphs of a TRAA with resolution $N = 5$.

learning bias introduced by accessing the materials with frequencies proportional to $\vec{x} = [q_{s(t)}, r_{s(t)}]$. Also observe that the overall learning scheme does not produce any absorbing states, and is, accordingly, ergodic supporting dynamic environments. The effect of these properties is analysed in the next subsection.

Finally, after the automaton has had the opportunity to change its state, it provides output to the EDF Scheduler. That is, it outputs the material amounts $\vec{x} = [q_{s(t+1)}, r_{s(t+1)}]$ that have been changed.

Earliest Deadline First (EDF) Scheduler: The Scheduler takes material amounts $\vec{x} = [x_1, \ldots, x_n]$ as its input (for the two-material case the input is $\vec{x} = [x_1, x_2]$). The purpose of the Scheduler is:

1. To provide accesses to the Stochastic Environment in a sequential manner, and
2. To make sure that the unit volume value functions are accessed with frequencies proportional to $\vec{x}$.

The reader should note that our scheme does not rely on accessing the unit volume value functions sequentially with frequencies proportional to $\vec{x}$ for solving the knapsack problem. However, this restriction is obviously essential for solving the problem *incrementally* and *on-line* (or rather in a "real-time" manner). Note that since it, in some cases, may be essential to access each unit volume value function with a constant period and not randomly (for example, in the earlier-alluded-to problem which analyzes web page polling), we use the Earliest Deadline First (EDF) Scheduling to access the functions according to $\vec{x}$.

3.1.2 Analysis of the TRAA Solution

In this section we characterize the optimal solution to a Stochastic NEFK Problem. Thereafter, we analyze the feedback connection of the TRAA and the Stochastic Environment — we prove that the TRAA is asymptotically optimal in the sense that it can find material allocations arbitrarily close to the solution of the Stochastic NEFK Problem.

Lemma 1. *The material mix $\vec{x} = [x_1, \ldots, x_n]$ is a solution to a given Stochastic NEFK Problem if (1) the derivatives of the expected material amount values are all equal at $\vec{x}$, (2) the mix fills the knapsack, and (3) every material amount*

is positive, i.e.:

$$f_1'(x_1) = \cdots = f_n'(x_n)$$
$$\sum_1^n x_i = c \textit{ and } \forall i \in \{1, \ldots, n\}, x_i \geq 0.$$

The above lemma is based on the well-known principle of Lagrange Multipliers [2, 5], and its proof is therefore omitted here for the sake of brevity. Instead, we will start by analyzing the *two-material* problem and the TRAA. Multiple TRAAs will then be organized in a hierarchy with the aim of tackling n*-material* problems.

For the two-material problem, let $\vec{x}^* = [x_1^*, x_2^*]$ denote a solution, as defined above. Note that since x_2^* can be obtained from x_1^*, we will concentrate on finding x_1^*.

Theorem 1. *The TRAA solution scheme specified by (1)–(3) is asymptotically optimal.*

Proof. Our aim is to prove that as the resolution, N, is increased indefinitely, the expected value of the TRAA output, $x_1(t)$, converges towards the solution of the problem, x_1^*, implying that:

$$\lim_{N\to\infty} \lim_{t\to\infty} E[x_1(t)] \to x_1^*.$$

We shall prove the above by analyzing the properties of the underlying Markov chain, which is specified by the EDF Scheduler, the rules (1)–(3) (the TRAA), and the Environment. As can be seen from (1)–(3), the states of the chain are the integers $j \in \{1, 2, \ldots, N\}$. In brief, rules (1)–(3), when interacting with the EDF Scheduler and the Environment, obey the Markov chain with transition matrix $H = [h_{ij}]$, where

$$h_{j,j-1} = r_j \cdot p_2(r_j) \cdot q_j, \quad 1 < j \leq N \tag{4}$$
$$h_{j,j+1} = q_j \cdot p_1(q_j) \cdot r_j, \quad 1 \leq j < N \tag{5}$$
$$h_{j,j} = 1 - h_{j,j-1} - h_{j,j+1}, \quad 1 < j < N, \tag{6}$$

and, accordingly,

$$h_{1,1} = 1 - h_{1,2} \tag{7}$$
$$h_{N,N} = 1 - h_{N,N-1}. \tag{8}$$

Clearly, H represents a single closed communicating class whose periodicity is unity. The chain is ergodic, and the limiting probability vector is given by the eigenvector of H^T corresponding to eigenvalue unity. Let this vector be $\Pi = [\pi_1, \pi_2, \ldots, \pi_N]$. Then, Π satisfies:

$$\begin{bmatrix} h_{1,1} & h_{1,2} & 0 & \cdot & \cdot & \cdot & \cdot & 0 \\ h_{2,1} & h_{2,2} & h_{2,3} & 0 & \cdot & \cdot & \cdot & 0 \\ 0 & h_{3,2} & h_{3,3} & h_{3,4} & 0 & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & \cdot & 0 & h_{N-2,N-3} & h_{N-2,N-2} & h_{N-2,N-1} & 0 \\ 0 & \cdot & \cdot & \cdot & 0 & h_{N-1,N-2} & h_{N-1,N-1} & h_{N-1,N} \\ 0 & \cdot & \cdot & \cdot & \cdot & 0 & h_{N,N-1} & h_{N,N} \end{bmatrix}^T \begin{bmatrix} \pi_1 \\ \pi_2 \\ \pi_3 \\ \cdot \\ \cdot \\ \pi_{N-2} \\ \pi_{N-1} \\ \pi_N \end{bmatrix} = \begin{bmatrix} \pi_1 \\ \pi_2 \\ \pi_3 \\ \cdot \\ \cdot \\ \pi_{N-2} \\ \pi_{N-1} \\ \pi_N \end{bmatrix} \tag{9}$$

The details of solving Equation (9) are quite cumbersome, and we undertake it now. Observe that our aim is to prove that the probability mass of Π lies arbitrarily close to the solution of the knapsack problem, $\vec{x}^* = [x_1^*, x_2^*]$, as N goes to infinity. Before we go through the fine details, we outline the proof strategy as follows. We first explicitly solve for the quantities $\{\pi_i\}$ by solving the underlying difference equations. We then define a function U that forms an upper bound for Π. We proceed to show that the upper bound goes to zero outside an arbitrarily close vicinity of x_1^*, as the resolution, N, goes to infinity. Accordingly, since Π is a probability distribution, and since U is its upper bound, increasing the resolution towards infinity moves the probability mass of Π arbitrarily close to x_1^*.

The details of the proof follow. Our first step is to reformulate the individual row-wise equations from the matrix Equation (9) recursively. Expanding the first row of Equation (9) yields:

$$\pi_1 \cdot h_{1,1} + \pi_2 \cdot h_{2,1} = \pi_1 \Rightarrow \pi_2 = \frac{(1 - h_{1,1}) \cdot \pi_1}{h_{2,1}} = \frac{h_{1,2}}{h_{2,1}} \cdot \pi_1. \tag{10}$$

Expanding the second row of Equation (9) and substituting Equation (10) yields:

$$\pi_1 \cdot h_{1,2} + \pi_2 \cdot h_{2,2} + \pi_3 \cdot h_{3,2} = \pi_2 \Rightarrow \pi_3 = \frac{h_{2,3}}{h_{3,2}} \cdot \pi_2. \tag{11}$$

Arguing in a similar way in a row-by-row manner, it can be seen[10] that

$$\pi_{k-1} = \frac{h_{k,k-1}}{h_{k-1,k}} \cdot \pi_k \tag{12}$$

for $0 < k \leq N$, which, on reversing the recursion, yields for $0 \leq k < N$,

$$\pi_{k+1} = \frac{h_{k,k+1}}{h_{k+1,k}} \cdot \pi_k. \tag{13}$$

Let $\alpha(x_1, N) = \left\lfloor \frac{x_1}{\frac{1}{N+1}} \right\rfloor$ and $\beta(x_1, N) = \left\lceil \frac{x_1}{\frac{1}{N+1}} \right\rceil$. Clearly, $\left[\frac{\alpha(x_1,N)}{N+1}, \frac{\beta(x_1,N)}{N+1} \right]$ is the interval that most accu-

[10] We omit the laborious algebraic steps in the interest of readability.

rately approximate x_1 given the resolution N. In particular, with $z = \alpha(x_1^*, N)$, the solution x_1^* is found in the interval $\left[\frac{z}{N+1}, \frac{z+1}{N+1}\right]$. The crucial part of our proof is to reformulate Π in terms of π_z and π_{z+1}, using (12)–(13). More specifically, for $j \in \{1, \ldots, z-1\}$ we have:

$$\pi_j \quad = \quad \pi_z \cdot \prod_{k=z}^{j+1} \frac{h_{k,k-1}}{h_{k-1,k}}. \tag{14}$$

Correspondingly, and arguing in an analogous manner, for $j \in \{z+2, \ldots, N\}$ we have:

$$\pi_j \quad = \quad \pi_{z+1} \cdot \prod_{k=z+1}^{j-1} \frac{h_{k,k+1}}{h_{k+1,k}}. \tag{15}$$

In other words, we represent Π in terms of two of its components: π_z and π_{z+1}.

We are now ready to define the upper bound U for Π:

$$U[i,z] = \begin{cases} \pi_z \cdot M^{z-i} & \text{if } i \leq z \\ \pi_{z+1} \cdot M^{i-(z+1)} & \text{if } i \geq z+1 \end{cases} \tag{16}$$

where:

$$M = \max\left[\max_{k \leq z}\left\{\frac{h_{k,k-1}}{h_{k-1,k}}\right\}, \max_{k \geq z+1}\left\{\frac{h_{k,k+1}}{h_{k+1,k}}\right\}\right]. \tag{17}$$

As seen, the definition of M clearly makes U an upper bound for Π.

Our final goal is to show that as the resolution N goes to infinity, U goes to zero outside an arbitrarily close vicinity of x_1^*:

$$\lim_{N \to \infty} \quad U[\alpha(x_1, N), \alpha(x_1^*, N)] \to 0 \quad \text{if } x_1 \neq x_1^* \tag{18}$$

We shall argue that the latter is guaranteed to happen if we have $0 < \frac{h_{k,k-1}}{h_{k-1,k}} < 1$ for $k \in \{2, \ldots, z\}$ and $0 < \frac{h_{k,k+1}}{h_{k+1,k}} < 1$ for $k \in \{z+1, \ldots, N-1\}$, because then we get $0 < M < 1$. We argue this by considering the equilibrium (asymptotic) value of $E[\pi(t)]$ for any finite N. This argument can be separated into three different cases as in [13]:

1. The first case is when $\frac{z}{N+1}$ is close to zero. In this case the maximum is quickly reached and then geometrically falls away.

2. When $\frac{z}{N+1}$ is close to 1, the value of π_i geometrically increases but when the maximum is reached it quickly falls away. For both these cases when $N \to \infty$, most of the probability mass will be centered in a small interval around z.

3. The third case is slightly more complex because it involves $\frac{z}{N+1}$ being away from either end. This case

must be broken down into two distinct geometric series, one representing the geometric series from π_1 to π_z and the other from π_{z+1} to π_N. The first series increases until it reaches the maximum at π_Z. The increase is geometric (or rather, exponential as $N \to \infty$), and the geometric ratio is bounded by the bound given by the quantity M above. The second series starts at the maximum at the value π_{z+1} and then decreases until π_N is reached. Again, the decrease is geometric (i.e., exponential as $N \to \infty$), and the geometric ratio is bounded by the quantity M above. In this case the probability mass will be centered within a small interval around $\frac{z}{N+1}$ and $\frac{z+1}{N}$ as $N \to \infty$ because of the law of the *sum* of the elements of a geometric series possessing a common ratio which is greater than unity.

First of all, since the difference between $\frac{k}{N+1}$ and $\frac{k-1}{N+1}$ goes to zero as N goes to infinity, and since $p_1(x)$ is continuous, we have:

$$\lim_{N\to\infty} \frac{h_{k,k-1}}{h_{k-1,k}} = \lim_{N\to\infty} \frac{r_k \cdot p_2(r_k) \cdot q_k}{q_k \cdot p_1(q_k) \cdot r_k} \tag{19}$$

$$= \lim_{N\to\infty} \frac{p_2(r_k)}{p_1(q_k)}. \tag{20}$$

Secondly, from Lemma 1 we can conclude that $p_1(q_k) > p_2(r_k)$ for $k \in \{2, \ldots, z\}$. Therefore, $0 < \frac{h_{k,k-1}}{h_{k-1,k}} < 1$ for $k \in \{2, \ldots, z\}$ as N goes to infinity.

Showing that we have $0 < \frac{h_{k,k+1}}{h_{k+1,k}} < 1$ for $k \in \{z+1, \ldots, N-1\}$ follows analogously, and the proof is left out here for the sake of brevity.

Accordingly, Π must go to zero outside an arbitrarily close vicinity of x_1^* as the resolution N goes to infinity. This, in turn, means that the probability mass of Π will lie arbitrarily close to x_1^*. In other words, the TRAA is asymptotically optimal. □

3.2 Details of the H-TRAA Solution

3.2.1 Design of the H-TRAA Solution

In this section we propose a hierarchical scheme for solving n-material problems. The scheme takes advantage of the TRAA's ability to solve two-material problems asymptotically, by organizing them hierarchically.

Construction of Hierarchy. The hierarchy of TRAAs, which we hereafter will refer to as H-TRAA, is constructed as follows[11]. First of all, the hierarchy is organized as a balanced binary tree with depth $D = \log_2(n)$. Each node in the hierarchy can be related to three entities: (1) a set of materials, (2) a partitioning of the material set into two subsets of equal size, and (3) a dedicated TRAA that allocates a given amount of resources among the two subsets.

[11] We assume that $n = 2^\gamma, \gamma \in \mathbb{N}^+$, for the sake of clarity. If the number of materials is less than this, we can assume the existence of additional materials whose values are "zero", and who thus are not able to contribute to the final optimal solution.

Root Node: The hierarchy root (at depth 1) is assigned the complete set of materials $S_{1,1} = \{1, \ldots, n\}$. These n materials are partitioned into two disjoint and exhaustive subsets of equal size: $S_{2,1}$ and $S_{2,2}$. An associated TRAA, $T_{1,1}$, decides how to divide the full knapsack capacity c (which, for the sake of notational correctness will be referred to as $c_{1,1}$) among the two subsets. That is, subset $S_{2,1}$ receives the capacity $c_{2,1}$ and subset $S_{2,2}$ receives the capacity $c_{2,2}$, with $c_{2,1} + c_{2,2} = c_{1,1}$. Accordingly, *this* TRAA is given the power to prioritize one subset of the materials at the expense of the other.

Nodes at Depth d: Node $j \in \{1, \ldots, 2^{d-1}\}$ at depth d (where $1 < d \leq D$) refers to: (1) the material subset $S_{d,j}$, (2) a partitioning of $S_{d,j}$ into the subsets $S_{d+1,2j-1}$ and $S_{d+1,2j}$, and (3) a dedicated TRAA, $T_{d,j}$. Observe that since level $D+1$ of the H-TRAA is non-existent, we use the convention that $S_{D+1,2j-1}$ and $S_{D+1,2j}$ refer to the primitive materials being processed by the leaf TRAA, $T_{D,j}$. Assume that the materials in $S_{d,j}$ has, as a set, been assigned the capacity $c_{d,j}$. The dedicated TRAA, then, decides how to allocate the assigned capacity $c_{d,j}$ among the subsets $S_{d+1,2j-1}$ and $S_{d+1,2j}$. That is, subset $S_{d+1,2j-1}$ receives the capacity $c_{d+1,2j-1}$ and subset $S_{d+1,2j}$ receives the capacity $c_{d+1,2j}$, with $c_{d+1,2j-1} + c_{d+1,2j} = c_{d,j}$

At depth D, then, each individual material can be separately assigned a fraction of the overall capacity by way of recursion, using the above allocation scheme.

Interaction of the H-TRAA with the EDF Scheduler and Environment. As in the single TRAA case, the H-TRAA interacts with an EDF Scheduler, which suggests which unit volume value function $p_i(x_i)$ to access next. A response is then generated from the Stochastic Environment using $p_i(x_i)$. This response is given to all the TRAAs that were involved in determining the material amount x_i, that is, the TRAAs in the hierarchy that have allocated capacity to a material subset that contains material i. Finally, a new candidate material mix $\vec{x} = [x_1, \ldots, x_n]$ is suggested by the H-TRAA to the EDF Scheduler.

Example I. Consider a 4-material problem. Figure 4 shows the associated hierarchy, constructed as described above. At the root level the TRAA $T_{1,1}$ divides the knapsack capacity among the two material subsets $\{1, 2\}$ and $\{3, 4\}$, respectively related to TRAA $T_{2,1}$ and $T_{2,2}$. At the level below, then, the TRAA $T_{2,1}$ allocates its share of the capacity among material 1 and material 2, while TRAA $T_{2,2}$ assigns *its* share of the capacity to material 3 and material 4. Based on the present assignment at time t, the EDF Scheduler selects material i, suggesting the amount $x_i(t)$ to the Stochastic Environment. The Stochastic Environment, in turn, responds with a randomly drawn material unit volume value, $v_i(t)$, using the probability value function $p_i(x_i)$. By way of example, if $i = 3$, the latter feedback is given to TRAAs $T_{1,1}$ and $T_{2,1}$, which update their states accordingly, and the feedback loop continues.

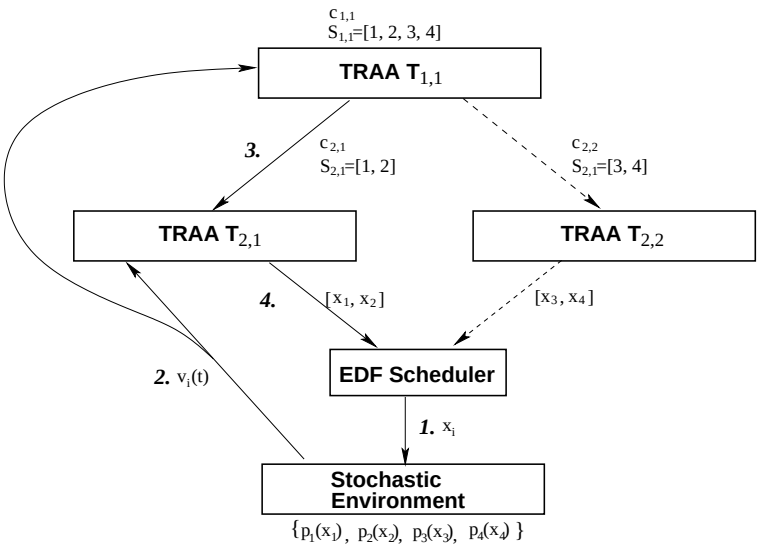

Figure 4: A Hierarchy of Twofold Resource Allocation Automata (H-TRAA) interacting with a Scheduler and an unknown Stochastic Environment as explained in Example I.

3.2.2 Analysis of the H-TRAA Solution

In the previous section we proved that an individual TRAA is asymptotically optimal. We will now consider the H-TRAA and prove its optimality. More specifically, we shall show that if each individual TRAA in the hierarchy has solved its own two-material problem, a solution to the complete n-material Knapsack Problem has also been produced.

Theorem 2. *Let $T_{d,j}$ be an arbitrary TRAA at level d of the H-TRAA associated with the node whose index is j. Then, if every single TRAA, $T_{d,j}$, in the H-TRAA has found a local solution with proportions $c_{d+1,2j-1}$ and $c_{d+1,2j}$ satisfying*

$$f'_{d+1,2j-1}(c_{d+1,2j-1}) = f'_{d+1,2j}(c_{d+1,2j}),$$

the overall Knapsack Problem involving n materials that are hierarchically placed in $\log_2 n$ levels of TRAAs, also attains the global optimum solution.

Proof. We intend to prove the above theorem by means of induction, using the hierarchical H-TRAA structure defined in the paragraph titled *Construction of Hierarchy*.

Basis: The *Basis* case concerns the nodes at the leaves, which, indeed, deal with the primitive materials themselves. Let a and b ($a, b \in \{1, \ldots, n\}$) be any two materials processed by a TRAA, $T_{D,u}$, at a leaf node (i.e., at depth D=$\log_2 n$) in the H-TRAA. The latter decides how to allocate an assigned capacity $c_{D,u}$ among the two materials a and b, with relative proportions x_a and x_b respectively. Observe that since a and b are the only two materials relevant to this TRAA, by virtue of the construction of the TRAA, $\frac{x_a}{x_a+x_b}$ and $\frac{x_b}{x_a+x_b}$ are the conditional probabilities of choosing a and b respectively, conditioned on the event that the knapsack

had only to be filled with these primitive materials. Since, by virtue of Theorem 1, we know that the TRAA will find a local solution $[x_a, x_b]$, the foundation of the solution determined by the Lagrangian yields:

$$f'_a(x_a) = f'_b(x_b) \Rightarrow f'_{D+1,2u-1}(c_{D+1,2u-1}) = f'_{D+1,2u}(c_{D+1,2u}), \text{ with } c_{D+1,2u-1} + c_{D+1,2u} = c_{D,u},$$

thus proving the basis of the induction.

Induction Step: Consider any interior-node TRAA $T_{d,j}$ whose index at depth d is j in the H-TRAA hierarchy. The TRAA associated with this node decides how to allocate an assigned capacity $c_{d,j}$ among two disjoint subsets $S_{d+1,2j-1} = \{\alpha_1, \ldots, \alpha_m\}$ and $S_{d+1,2j} = \{\beta_1, \ldots, \beta_m\}$ of *composite* materials, where each α_i and β_i is, in itself, a primitive material. To simplify notation, let $\vec{\alpha} = \{\alpha_1, \ldots, \alpha_m\}$ and $\vec{\beta} = \{\beta_1, \ldots, \beta_m\}$. Observe that the union of the sets $\vec{\alpha}$ and $\vec{\beta}$ is the input to the present TRAA, and the task of *this* TRAA is to assign the current knapsack capacity, $c_{d,j}$, so as to satisfying the Lagrangian solution for these two mutually exclusive and exhaustive subsets. Let $T_{d,j}$ assign the relative proportions to $\vec{\alpha}$ and $\vec{\beta}$ by the quantities $x_{\vec{\alpha}}$ and $x_{\vec{\beta}}$. Observe that since $\vec{\alpha}$ and $\vec{\beta}$ are the only two materials[12] relevant to this TRAA, by virtue of the construction of the TRAA, $\frac{x_{\vec{\alpha}}}{x_{\vec{\alpha}}+x_{\vec{\beta}}}$ and $\frac{x_{\vec{\beta}}}{x_{\vec{\alpha}}+x_{\vec{\beta}}}$ are the conditional probabilities of choosing $\vec{\alpha}$ and $\vec{\beta}$ respectively, conditioned on the event that the knapsack had only to be filled with these composite materials $\vec{\alpha}$ and $\vec{\beta}$. The solution to this TRAA will thus satisfy:

$$f'_{\vec{\alpha}}(x_{\vec{\alpha}}) = f'_{\vec{\beta}}(x_{\vec{\beta}}) \text{ where,} \tag{21}$$

$$f'_{\vec{\alpha}}(x_{\vec{\alpha}}) = \sum_{\alpha_i \in \vec{\alpha}} \frac{x_{\alpha_i}}{\sum_{\alpha_j \in \vec{\alpha}} x_{\alpha_j}} f'_{\alpha_i}(x_{\alpha_i}) \text{ and} \tag{22}$$

$$f'_{\vec{\beta}}(x_{\vec{\beta}}) = \sum_{\beta_i \in \vec{\beta}} \frac{x_{\beta_i}}{\sum_{\beta_j \in \vec{\beta}} x_{\beta_j}} f'_{\beta_i}(x_{\beta_i}). \tag{23}$$

Since each α_i and β_i is a primitive material, and we are working our way *up* the H-TRAA hierarchy, we can invoke the inductive hypothesis to relate x_{α_i} and x_{β_i} for all i. By virtue of the inductive hypothesis and the Lagrangian solution at *every* level up the H-TRAA till level d, we know that for both of the material subsets $S_{d+1,2j-1}$ and $S_{d+1,2j}$ the following are true:

$$f'_{\alpha_1}(x_{\alpha_1}) = \cdots = f'_{\alpha_m}(x_{\alpha_m}) \tag{24}$$

$$f'_{\beta_1}(x_{\beta_1}) = \cdots = f'_{\beta_m}(x_{\beta_m}). \tag{25}$$

To simplify the notation, let each of the quantities in Equation (24) equal $f'_{\alpha}(x_{\alpha})$, and each of the quantities in Equation (25) equal $f'_{\beta}(x_{\beta})$.

Substituting Equations (24) and (25) (which represent the induction hypothesis) into Equations (22) and

[12]The fact hat these are composite materials is irrelevant to the present TRAA. It merely treats $\vec{\alpha}$ and $\vec{\beta}$ as individual materials.

(23), the latter become:

$$f'_{\vec{\alpha}}(x_{\vec{\alpha}}) = f'_{\alpha}(x_{\alpha}) \sum_{\alpha_i \in \vec{\alpha}} \frac{x_{\alpha_i}}{\sum_{\alpha_j \in \vec{\alpha}} x_{\alpha_j}} \text{ and} \tag{26}$$

$$f'_{\vec{\beta}}(x_{\vec{\beta}}) = f'_{\beta}(x_{\beta}) \sum_{\beta_i \in \vec{\beta}} \frac{x_{\beta_i}}{\sum_{\beta_j \in \vec{\beta}} x_{\beta_j}}. \tag{27}$$

The summations on the RHSs of both of the Equations (26) and (27) can be trivially seen to sum to unity since they represent probabilities (in the conditioned spaces), implying that:

$$\forall i : f'_{\vec{\alpha}}(x_{\vec{\alpha}}) = f'_{\alpha_i}(x_{\alpha_i}) \text{ and} \tag{28}$$

$$\forall i : f'_{\vec{\beta}}(x_{\vec{\beta}}) = f'_{\beta_i}(x_{\beta_i}). \tag{29}$$

Combining the above with Equation (21) yields:

$$f'_{\alpha_1}(x_{\alpha_1}) = \cdots = f'_{\alpha_m}(x_{\alpha_m}) = f'_{\beta_1}(x_{\beta_1}) = \cdots = f'_{\beta_m}(x_{\beta_m}). \tag{30}$$

implying that the global optimum required by the Lagrangian has been found. Hence the theorem! □

Remarks: Theorem 2 has some very interesting consequences listed below:

1. The proof of Theorem 2 has tacitly assumed that all the automata have converged before the global convergence can be asserted. This implies that the TRAA $T_{d,j}$ is aware of its capacity, and that this is a known quantity to the TRAAs $T_{d+1,2j-1}$ and $T_{d+1,2j}$. In other words, if all the individual TRAAs converge to their local optimum, Theorem 2 states that the global optimum is attained. Conceptually, this can pose a small implementation-related problem. The fact is that the TRAAs of the lower level are converging even while the TRAA at the higher level is attempting to find *its* capacity. Therefore, essentially, the lower level TRAAs are working in a non-stationary environment. The strategy by which we can resolve this is to ensure that the higher level automata converge at a slower rate than the lower ones (thus guaranteeing a certain level of stationarity). In practice, however, we have observed that if the resolution parameter N is large enough (in the order of hundreds) the time varying phenomenon is marginal, and the TRAAs at all the levels tend to converge simultaneously.

2. Theorem 2 claims that the solution obtained by the convergence of the individual TRAAs leads to the global convergence of the overall optimization problem. But this claim means that the ordering of the materials at the leaf nodes does not carry any significance. This is, indeed, true! It turns out that if the nodes at the leaves are ordered in such a way that "more precious materials" lie in the same sub-tree, the weight associated with the sub-tree of the composite material containing these "more precious materials" will have a much larger weight, and the weight of the other sub-trees will be much smaller. As opposed to this, if the "more precious materials" lie in distinct sub-trees, the weights associated

with the respective sub-trees will be correspondingly compensated for.

4 Experimental Results: Optimal Polling Frequency Determination

4.1 Problem Background

Having obtained a formal solution to the model in which we set the NEFK, we shall now demonstrate how we can utilize this solution for the current problem being studied, namely, the optimal web-polling problem.

As mentioned in Section 1, in our work, we will denote the update detection probability of a web page i as d_i. Under the above conditions, d_i depends on the frequency, x_i, that the page is polled with, and is modeled using the following expression:

$$d_i(x_i) = 1 - q_i^{\frac{1}{x_i}}.$$

By way of example, consider the scenario that a web page remains unchanged in any single time step with probability 0.5. Then polling the web page uncovers new information with probability $1 - 0.5^3 = 0.875$ if the web page is polled every 3^{rd} time step (i.e., with frequency $\frac{1}{3}$) and $1 - 0.5^2 = 0.75$ if the web page is polled every 2^{nd} time step. As seen, increasing the polling frequency reduces the probability of discovering new information on each polling.

Given the above considerations, our aim is to find the page polling frequencies $\vec{x}$ that maximize the expected number of pollings uncovering new information per time step:

$$\begin{array}{ll} \text{maximize} & \sum_1^n x_i \times d_i(x_i) \\ \text{subject to} & \sum_1^n x_i = c \text{ and } \forall i = 1, \ldots, n, x_i \geq 0. \end{array}$$

4.2 H-TRAA Solution

In order to find a H-TRAA Solution to the above problem we must define the Stochastic Environment that the LA are to interact with. As seen in Section 3, the Stochastic Environment consists of the unit volume value functions $\mathcal{F}' = \{f'_1(x_1), f'_2(x_2), \ldots, f'_n(x_n)\}$, which are unknown to H-TRAA. We identify the nature of these functions by applying the principle of Lagrange multipliers to the above maximization problem. In short, after some simplification, it can be seen that the following conditions characterize the optimal solution:

$$\begin{array}{l} d_1(x_1) = d_2(x_2) = \cdots = d_n(x_n) \\ \sum_1^n x_i = c \text{ and } \forall i = 1, \ldots, n, x_i \geq 0 \\ . \end{array}$$

Since we are not able to observe $d_i(x_i)$ or q_i directly, we base our definition of $\mathcal{F}'$ on the result of polling web pages. Briefly stated, we want $f'_i(x_i)$ to instantiate to the value 0 with probability $1 - d_i(x_i)$ and to the

value 1 with probability $d_i(x_i)$. Accordingly, if the web page i is polled and i has been updated since our last polling, then we consider $f_i'(x_i)$ to have been instantiated to 1. And, if the web page i is unchanged, we consider $f_i'(x_i)$ to have been instantiated to 0.

4.3 Empirical Results

4.3.1 Experimental Setup

In this section we evaluate our learning scheme by comparing it with by comparing it with four classical policies using synthetic data. We have implemented the following classical policies, non of which invokes the Knapsack Problem.

Uniform: The uniform policy allocates monitoring resources uniformly across all web pages. This is the only classical policy of the four that can be applied directly in an unknown environment.

Proportional: In the proportional policy, the allocation of monitoring resources to web pages is proportional to the update frequencies of the web pages. Accordingly, this policy requires that the web page update frequencies are known.

Estimator: The estimator policy handles unknown web update frequencies by polling web pages *uniformly* in a parameter estimation phase, with the purpose of estimating update frequencies. After the parameter estimation phase, the proportional policy is applied, however, based on the estimated frequencies.

Optimal: This policy requires that update frequencies are known, and finds the optimal solution based on the principle of Lagrange multipliers [14, 21].

To evaluate web resource allocation policies, recent research advocates Zipf-like distributions [22] to generate realistic web page update frequencies [14, 21]. The Zipf distribution can be stated as follows [20]:

$$Z(k; s, N) = \frac{1/k^s}{\sum_{n=1}^{N} 1/n^s}$$

where N is the number of elements, k is their rank, and s is a parameter that governs the skewed-ness of the distribution (e.g., for $s = 0$ the distribution is uniform).

For our experiments, web pages are considered ranked according to their update frequencies, and the update probability of a web page is calculated from its rank. We use the following function to determine the update probability of each web page:

$$q_k(\alpha, \beta) = \frac{\alpha}{k^\beta}.$$

In this case, k refers to the web page of rank k and the parameter β determines the skewed-ness of the distribution, while $\alpha \in [0.0, 1.0]$ represents the magnitude of the update probabilities (i.e., the web page of rank 1 is updated with probability α each time step).

Without loss of generality, we normalize the web page polling capacity in our experiments to 1.0 poll per time step, and accordingly, we vary the average total number of web page updates per time step instead.

As we will see in the following, it turns that one of the strengths of the H-TRAA is its ability to take advantage of so-called spatial dependencies among materials. As mentioned earlier, in the above experimental setup, materials are spatially related in the sense that the updating probabilities decreases with the rank-index k. In order to starve the H-TRAA from this information, we opted to perturb this spatial structure. Each perturbation swapped the updating probabilities of a randomly selected material and the material succeeding it in the ranking. Based on the above, we conducted our experiments with 10^3, 10^4, 10^5 and 10^6 perturbations.

The results of our experiments are truly conclusive and confirm the power of the H-TRAA. Although several experiments were conducted using various α, β, and number of automata, we report for the sake of brevity mainly the results for 512 web pages (the main case from [14]) within the following environments:

- $\alpha = 0.3$; $\beta = 1.5$, where the average number of updates per time step is 0.76 and accordingly, below the web page polling capacity. The web page update distribution is highly skewed, as explored in [14].
- $\alpha = 0.3$; $\beta = 1.0$, where the average number of updates per time step is increased to 2.0 (twice the web page polling capacity) by making the web page update distribution less skewed (the normal Zipf distribution).
- $\alpha = 0.9$; $\beta = 1.5$, where the average number of updates is set to 2.3 by increasing the web page update probability. Because of the high values of both α and β, this environment turns out to be the most challenging one, discriminating clearly between the optimal policy and the proportional policy.

For these values, an ensemble of several independent replications with different random number streams was performed to minimize the variance of the reported results.

4.3.2 Configuring the H-TRAA

The H-TRAA can be configured by various means. First of all, the material amount space $(0, 1)$ need not be discretized uniformly. Instead, a nonlinear material amount space can be formed, as done for the LAKG in [8]. Furthermore, the discretization resolution N must also be set for each TRAA, possibly varying from TRAA to TRAA in the hierarchy. In short, the performance achieved for a particular problem can be optimized using these different means of configuring the H-TRAA. In this section, however, our goal is to evaluate the overall performance of the H-TRAA, without fine tuning. Therefore, we will only use a linear material amount space, as specified in Section 3. Furthermore, we will use the same resolution $N = 500$ for all the TRAAs in the hierarchy, independent of the specific knapsack problem at hand. Thus, our aim is to ensure a fair comparison with the present state of the art, namely, the LAKG scheme.

While the focus of the previous section was on learning only from material units of value 1 (rewards), with some simple modifications the H-TRAA scheme clearly supports the three well-established updating approaches:

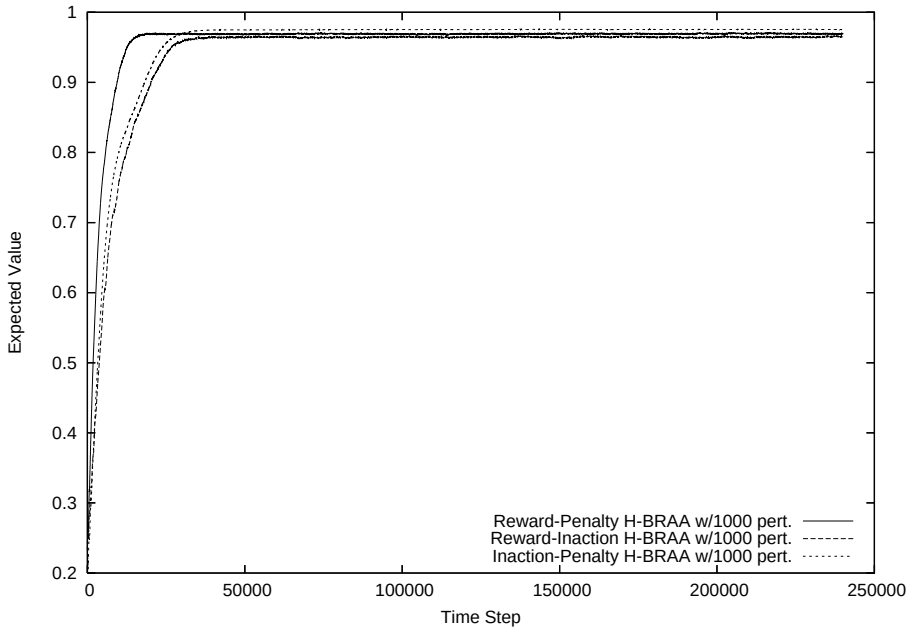

Figure 5: Using the $(\alpha = 0.9, \beta = 1.5)$-environment we observe that the *Inaction-Penalty* updating is the most accurate. However, the *Reward-Penalty* updating converges more quickly.

1. **Reward-Inaction**: In this case, the H-TRAA updates its state only when a material unit volume value of '1' is given as the feedback from the Stochastic Environment, which is the case studied in the previous section.

2. **Inaction-Penalty**: In this case, the H-TRAA updates its state only when a material unit volume value of '0' is given as feedback. Here, the reader will observe that the state transitions of the individual TRAAs from Section 3 are inverted.

3. **Reward-Penalty**: In this case, the H-TRAA updates its state in both of the above cases.

When exposed to the $(\alpha = 0.9, \beta = 1.5)$-environment, we see from Figure 5 that the *Inaction-Penalty* updating is the most accurate of the three approaches. However, the *Reward-Penalty* updating converges more quickly, since the state is updated both on rewards and on penalties. Only relying on rewards is slightly inferior to the other two approaches for $(\alpha = 0.9, \beta = 1.5)$-environment. Because we emphasize speed of learning in this paper, we will, in the following, only use *Reward-Penalty* updating. But we note, however, that the two other approaches produce similar results.

4.3.3 Static Environments

We see from Figure 6 that the proportional policy and the optimal policy provide more-or-less the same solution — a solution superior to the uniform policy solution. We also observe that the performance of the estimator scheme increases steadily with the length of the parameter estimation phase. The figure also shows the performance of the H-TRAA increases significantly quicker than the LAKG and the Estimator

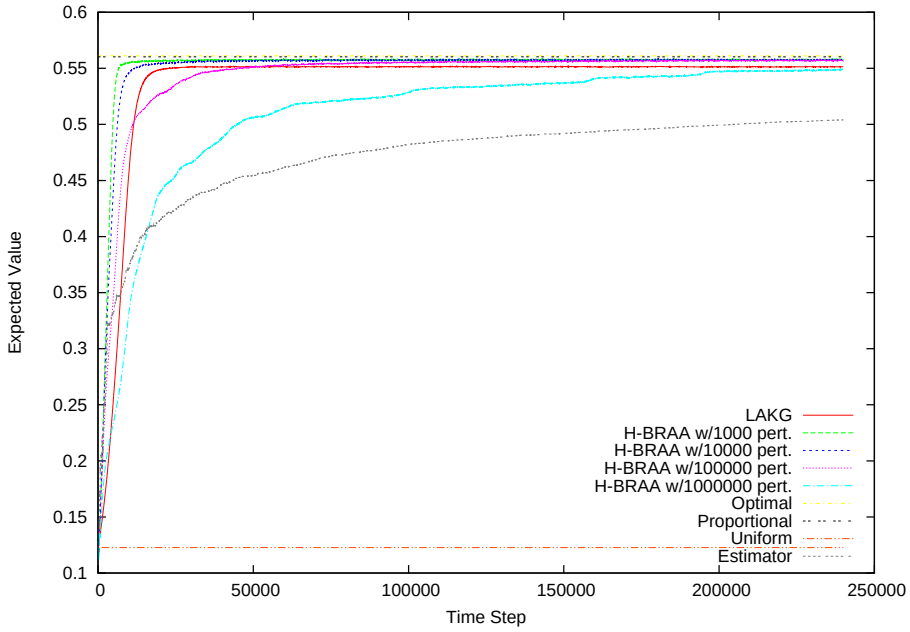

Figure 6: In the $(\alpha = 0.3, \beta = 1.5)$-environment, the H-TRAA scheme is superior to the LAKG scheme and the estimator scheme. However, for highly unstructured environments, the LAKG provides better performance.

schemes. However, when increasing the number of perturbations, the performance of the H-TRAA is reduced. Indeed, with $1,000,000$ perturbations, the LAKG turns out to converge both more quickly and more accurately than the H-TRAA. Note that even with $1,000,000$ perturbations, the H-TRAA provides performance equal to the LAKG if each TRAA in the hierarchy is given a resolution N that is twice as large as the resolution applied by any of its children. However, then the performance advantage of the H-TRAA is lost for the less perturbed cases. In this sense, the H-TRAA is more flexible than the LAKG, performing either better or similarly when the H-TRAA configuration is optimized for the problem at hand. Note that, in contrast to the Estimator scheme, the performance of both the H-TRAA and the LAKG is improved online (in a real-time manner) *without* invoking any parameter estimation phase.

As seen in Figure 7, a less skewed web page update distribution function makes the uniform policy more successful, mainly because a larger number of web pages will have a significant probability of being updated. For the same reason, the estimator scheme is able to lead to an improved performance quicker. In spite of this, the H-TRAA yields a superior performance.

The most difficult class of environments we simulate is an environment with a highly skewed web page update distribution ($\beta = 1.5$) combined with a high update probability ($\alpha = 0.9$). In such an environment, the optimal policy performs significantly better than the proportional policy, and so any scheme that converge towards a proportional policy solution will not attain an optimal performance. As seen in Figure 8, both the LAKG and the H-TRAA breach the performance boundary set by the proportional policy, and converges towards near-optimal solutions. The H-TRAA converges slightly quicker compared to the LAKG.

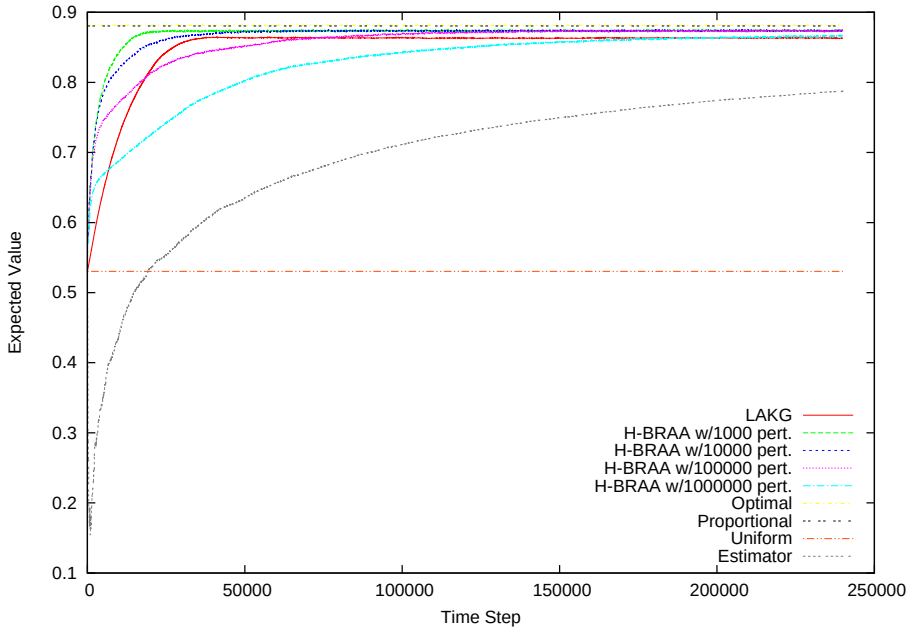

Figure 7: In the $(\alpha = 0.3, \beta = 1.0)$-environment, a less skewed web page update distribution makes the uniform policy as well as the estimator policy more successful, mainly because of more widely and abundant updating of the web pages.

4.3.4 Dynamic Environments

A dynamically changing environment is particularly challenging because the optimal solution is time dependent. In such cases, the current resource allocation solution should be modified according to the environmental changes. When, additionally, the environment and its characteristics are unknown, any changes must first be learned before any meaningful modification can take place.

In order to simulate a dynamic environment, we change the ranking of the web pages at every r^{th} web page poll — a single web page is selected by sampling from the current Zipf-distribution, and this web page switches rank with the succeeding web page in the ranking. As a result, the Zipf-distribution also changes. This means that the web monitor is allowed to conduct r web page polls before the environment changes. Figure 9 demonstrates the ability of our scheme to re-learn in a switching environment for $r = 80,000$. As seen in the figure, the H-TRAA quickly recovers after the environment has changed, and then moves towards a new near optimal solution. Also, the H-TRAA clearly outperforms the LAKG.

In the previous dynamic environment, the H-TRAA was able to fully recover to a near-optimal solution because of the low frequency of environmental changes. Figure 10 demonstrates the behavior of the automata in a case when this frequency is increased to $r = 1,000$. As seen, the automata still quickly and steadily improve the initial solution, but are obviously never allowed to reach an optimal solution. The reader should however note how the quickly-changing environment is not able to hinder the automata from stabilizing on a solution superior to the solutions found by the estimator scheme. Again, the H-TRAA performs better than the LAKG.

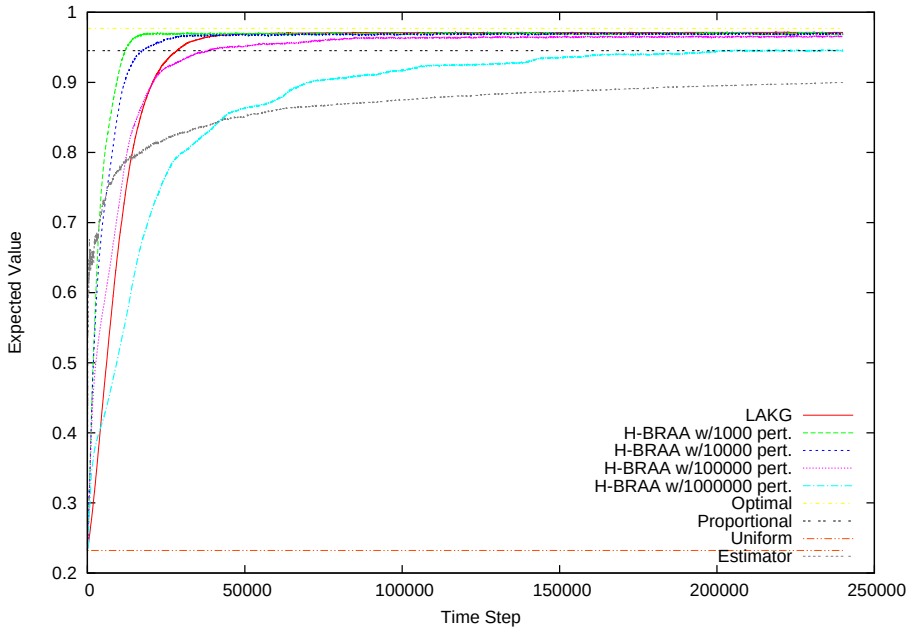

Figure 8: This figure shows that in the $(\alpha = 0.9, \beta = 1.5)$-environment, the H-TRAA scheme breaches the performance boundary set by the proportional policy, converging towards near-optimal solutions.

Clearly, these results demonstrate how the H-TRAA can perform when the environment is switching with a fixed period (in this case $r = 80000$ and $r = 1000$). However, we believe that similar results will be obtained if r is not fixed, but changing in such a way that the scheme has enough time to learn the parameters of the updated environment.

4.3.5 Scalability

One of the motivations for designing the H-TRAA was to obtain improved scalability by means of hierarchical learning. As seen in Figure 11, extending the number of materials significantly increases the convergence time of the LAKG. An increased initial learning phase may be unproblematic in cases where the system will run correspondingly longer, adapting to less dramatic changes as they occur. However, as also seen from the figure, the adaptation speed increases with the number of materials too, when the LAKG is used.

The H-TRAA, however, is far less affected by the number of materials. In Figure 12 we observe that the initial learning phase is orders of magnitude faster than what can be achieved with the LAKG. Furthermore, the impact on adaptation speed is negligible!

5 Conclusions and Further Work

In this paper, we have considered the optimal web polling problem, which involves determining a strategy for monitoring the world wide web. The problem consists of repeatedly polling a selection of web pages so that changes that occur over time are detected. In particular, we have considered the case where we are

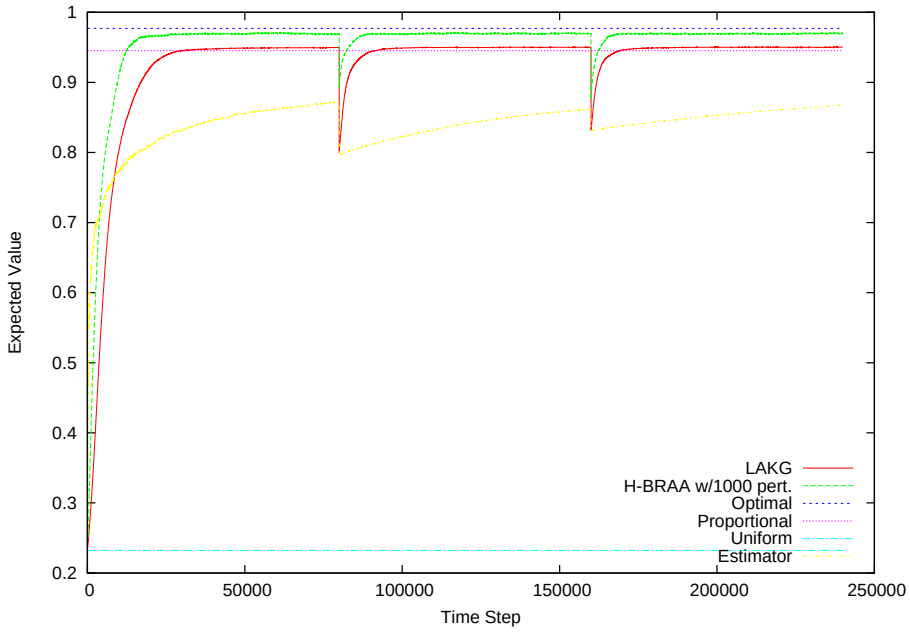

Figure 9: This figure shows the performance of the schemes n the $(\alpha = 0.9, \beta = 1.5)$-environment where the web page ranking changes at every $80,000^{th}$ web page poll. The H-TRAA finds near-optimal solutions initially, and recovers quickly after the respective environmental changes.

constrained to poll a *maximum* number of web pages per unit of time. This realistic constraint is typically dictated by the governing communication bandwidth, and by the speed limitations associated with the processing. Since only a fraction of the web pages can be polled within a given unit of time, our problem has been that of determining which web pages are to be polled, and we have attempted to do this in a manner that maximizes the number of changes detected is a reasonable choice. To solve the problem, we first modelled it as a *Stochastic Non-linear Fractional Knapsack Problem*. We then presented a completely new on-line Learning Automata (LA) system, namely, the *Hierarchy of Twofold Resource Allocation Automata* (H-TRAA), whose primitive component is a *Twofold Resource Allocation Automaton* (TRAA). Both the TRAA and the H-TRAA have been proven to be asymptotically optimal.

Comprehensive experimental results demonstrated that performance of the H-TRAA is superior to previous state-of-the-art schemes, and in particular, to a previously reported strategy which solves the same problem, the LAKG. We have also demonstrated that the H-TRAA scheme adapts to switching web distribution functions, allowing us to operate in dynamic environments. Finally, we have also provided empirical evidence to show that the H-TRAAs possess a sub-linear scaling property.

In our further work, we aim to develop alternate LA-based solutions for different classes of knapsack problems, including the NP-hard integer knapsack problem, which we hope to then apply to the www. Indeed, we are also currently investigating how other classes of LA can form the basis for novel knapsack-based learning problems.

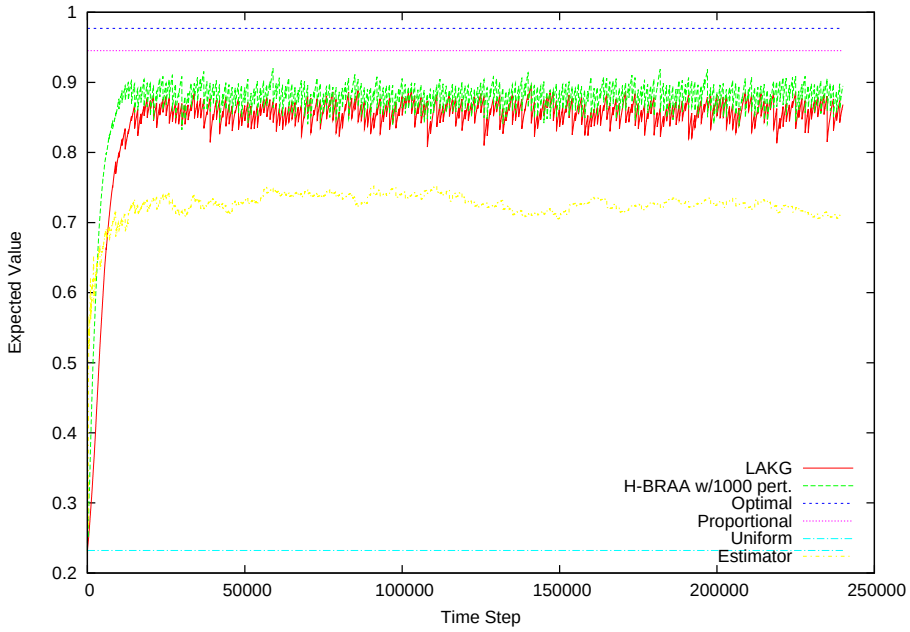

Figure 10: In the $(\alpha = 0.9, \beta = 1.5)$-environment where the web page ranking changes every $1,000^{th}$ poll. Observe that the H-TRAA is able to steadily improve the initial solution, but is never allowed to reach an optimal solution due to the nature of the switching.

References

[1] Paul E. Black. Fractional knapsack problem. *Dictionary of Algorithms and Data Structures*, 2004.

[2] K. M. Bretthauer and B. Shetty. The Nonlinear Knapsack Problem — Algorithms and Applications. *European Journal of Operational Research*, 138:459–472, 2002.

[3] C. Castillo. *Effective Web Crawling*. PhD thesis, University of Chile, 2004.

[4] B. C. Dean, M. X. Goemans, and J. Vondrdk. Approximating the Stochastic Knapsack Problem: the benefit of adaptivity. In *45th Annual IEEE Symposium on Foundations of Computer Science*, pages 208–217. IEEE, 2004.

[5] B. Fox. Discrete optimization via marginal analysis. *Management Sciences*, 13(3):211–216, 1966.

[6] O.-C. Granmo and B. J. Oommen. On Allocating Limited Sampling Resources Using a Learning Automata-based Solution to the Fractional Knapsack Problem. In *Proceedings of the 2006 International Intelligent Information Processing and Web Mining Conference (IIS:IIPW'06)*, Advances in Soft Computing, pages 263–272. Springer, 2006.

[7] O.-C. Granmo, B. J. Oommen, S. A. Myrer, and M. G. Olsen. Determining Optimal Polling Frequency Using a Learning Automata-based Solution to the Fractional Knapsack Problem. In *Proceedings of the*

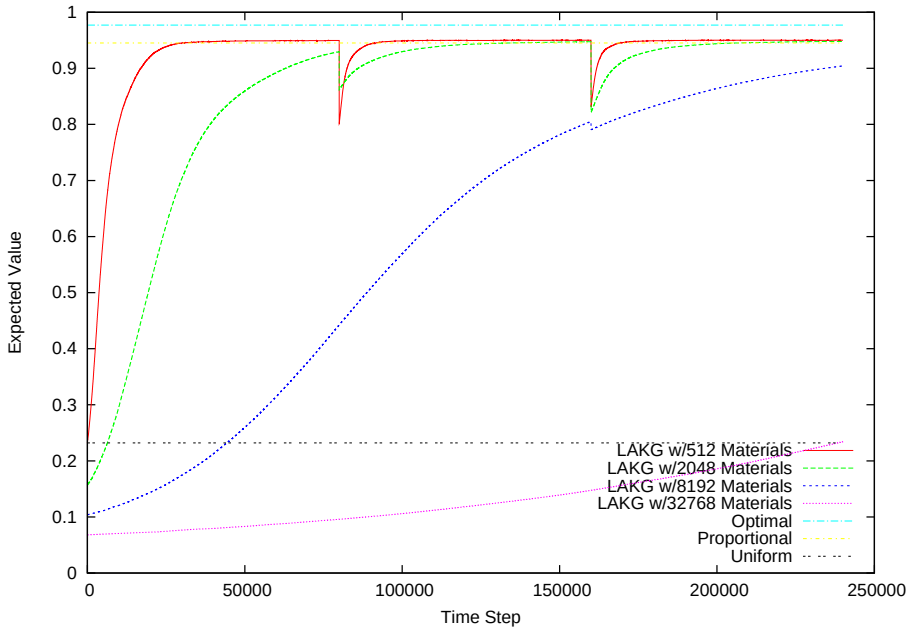

Figure 11: From this figure we see that extending the number of materials significantly increases the convergence and adaption time of the LAKG.

2006 IEEE International Conferences on Cybernetics & Intelligent Systems (CIS) and Robotics, Automation & Mechatronics (RAM), pages 73–79. IEEE, 2006.

[8] O.-C. Granmo, B. J. Oommen, S. A. Myrer, and M. G. Olsen. Learning Automata-based Solutions to the Nonlinear Fractional Knapsack Problem with Applications to Optimal Resource Allocation. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 37(1):166–175, 2007.

[9] O.-C. Granmo and B. John Oommen. Solving Stochastic Nonlinear Resource Allocation Problems Using a Hierarchy of Twofold Resource Allocation Automata. *IEEE Transactions on Computers*, 59(4):545–560, 2010.

[10] H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack Problems*. Springer, 2004.

[11] K. S. Narendra and M. A. L. Thathachar. *Learning Automata: An Introduction*. Prentice Hall, 1989.

[12] B. J. Oommen. Absorbing and Ergodic Discretized Two Action Learning Automata. *IEEE Transactions on Systems, Man and Cybernetics*, 16:282–293, 1986.

[13] B. J. Oommen. Stochastic Searching on the Line and its Applications to Parameter Learning in Nonlinear Optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 27(4):733–739, 1997.

[14] S. Pandey, K. Ramamritham, and S. Chakrabarti. Monitoring the Dynamic Web to Respond to Continuous Queries. In *12th International World Wide Web Conference*, pages 659–668. ACM Press, 2003.

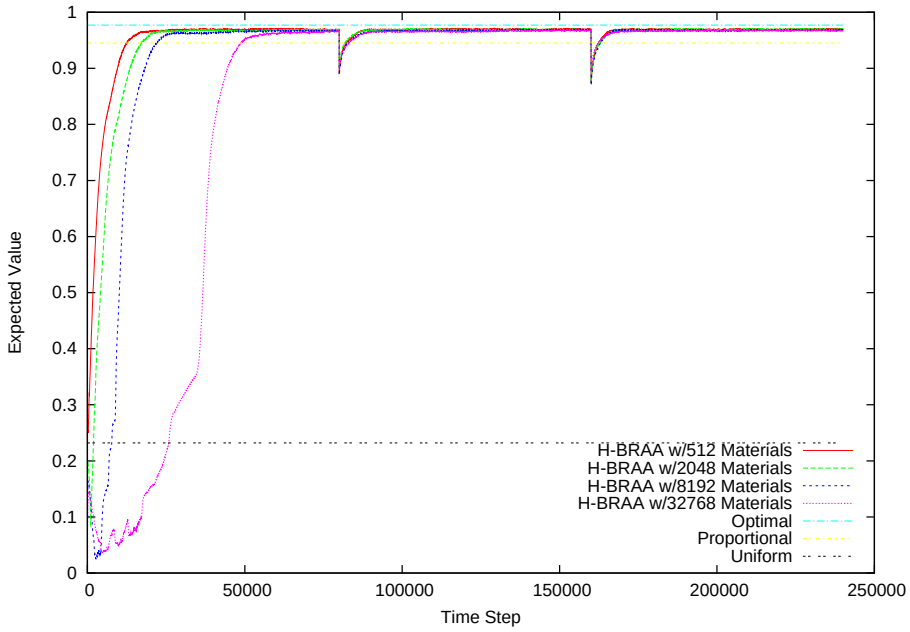

Figure 12: From this figure we see that the H-TRAA scales sub-linearly with the number of materials, and that the impact on the adaptation speed is negligible.

[15] K. W. Ross and D.H.K. Tsang. The stochastic knapsack problem. *IEEE Transactions on Communications*, 37(7), 1989.

[16] M. Snaprud, N. Ulltveit-Moe, O.-C. Granmo, M. Rafoshei-Klev, A. Wiklund, and A. Sawicka. Quantitative Assessment of Public Web Sites Accessibility - Some Early Results. In *The Accessibility for All Conference*, 2003.

[17] E. Steinberg and M. S. Parks. A Preference Order Dynamic Program for a Knapsack Problem with Stochastic Rewards. *The Journal of the Operational Research Society*, 30(2):141–147, 1979.

[18] M. A. L. Thathachar and P. S. Sastry. *Networks of Learning Automata: Techniques for Online Stochastic Optimization*. Kluwer Academic Publishers, 2004.

[19] M. L. Tsetlin. *Automaton Theory and Modeling of Biological Systems*. Academic Press, 1973.

[20] Wikipedia. Zipf's law — wikipedia, the free encyclopedia, 2006. [Online; accessed 5-December-2006].

[21] J. L. Wolf, M. S. Squillante, J. Sethuraman, and K. Ozsen. Optimal Crawling Strategies for Web Search Engines. In *11th International World Wide Web Conference*, pages 136–147. ACM Press, 2002.

[22] G. Zipf. *Human Behavior and the Principle of Least Effort*. Addison-Wesley, 1949.